Postel

Network Working Group
Request for Comments: 1049

M. Sirbu
CMU
March 1988

A CONTENT-TYPE HEADER FIELD FOR INTERNET MESSAGES

STATUS OF THIS MEMO

This RFC suggests proposed additions to the Internet Mail Protocol, RFC-822, for the Internet community, and requests discussion and suggestions for improvements. Distribution of this memo is unlimited.

ABSTRACT

A standardized Content-type field allows mail reading systems to automatically identify the type of a structured message body and to process it for display accordingly. The structured message body must still conform to the RFC-822 requirements concerning allowable characters. A mail reading system need not take any specific action upon receiving a message with a valid Content-Type header field. The ability to recognize this field and invoke the appropriate display process accordingly will, however, improve the readability of messages, and allow the exchange of messages containing mathematical symbols, or foreign language characters.

Table of Contents

1. Introduction 1
2. Problems with Structured Messages 3
3. The Content-type Header Field 5
   3.1. Type Values 5
   3.2. Version Number 6
   3.3. Resource Reference 6
   3.4. Comment. 7
4. Conclusion 7

1. Introduction

As defined in RFC-822, [2], an electronic mail message consists of a number of defined header fields, some containing structured information (e.g., date, addresses), and a message body consisting of an unstructured string of ASCII characters.

The success of the Internet mail system has led to a desire to use the mail system for sending around information with a greater degree of structure, while remaining within the constraints imposed by the limited character set. A prime example is the use of mail to send a

document with embedded TROFF formatting commands. A more sophisticated example would be a message body encoded in a Page Description Language (PDL) such as Postscript. In both cases, simply mapping the ASCII characters to the screen or printer in the usual fashion will not render the document image intended by the sender; an additional processing step is required to produce an image of the message text on a display device or a piece of paper.

In both of these examples, the message body contains only the legal character set, but the content has a structure which produces some desirable result after appropriate processing by the recipient. If a message header field could be used to indicate the structuring technique used in the message body, then a sophisticated mail system could use such a field to automatically invoke the appropriate processing of the message body. For example, a header field which indicated that the message body was encoded using Postscript could be used to direct a mail system running under Sun Microsystem's NEWS window manager to process the Postscript to produce the appropriate page image on the screen.

Private header fields (beginning with "X-") are already being used by some systems to affect such a result (e.g., the Andrew Message System developed at Carnegie Mellon University). However, the widespread use of such techniques will require general agreement on the name and allowed parameter values for a header field to be used for this purpose.

We propose that a new header field, "Content-type:" be recognized as the standard field for indicating the structure of the message body. The contents of the "Content-Type:" field are parameters which specify what type of structure is used in the message body.

Note that we are not proposing that the message body contain anything other than ASCII characters as specified in RFC-822. Whatever structuring is contained in the message body must be represented using only the allowed ASCII characters. Thus, this proposal should have no impact on existing mailers, only on mail reading systems.

At the same time, this restriction eliminates the use of more general structuring techniques such as Abstract Syntax Notation, (CCITT Recommendation X.409) as used in the X.400 messaging standard, which are octet-oriented.

This is not the first proposal for structuring message bodies. RFC-767 discusses a proposed technique for structuring multi-media mail messages. We are also aware that many users already employ mail to send TROFF, SCRIBE, TEX, Postscript or other structured information. Such postprocessing as is required must be invoked

manually by the message recipient who looks at the message text displayed as conventional ASCII and recognizes that it is structured in some way that requires additional processing to be properly rendered. Our proposal is designed to facilitate automatic processing of messages by a mail reading system.

2. Problems with Structured Messages

Once we introduce the notion that a message body might require some processing other than simply painting the characters to the screen we raise a number of fundamental questions. These generally arise due to the certainty that some receiving systems will have the facilities to process the received message and some will not. The problem is what to do in the presence of systems with different levels of capability.

First, we must recognize that the purpose of structured messages is to be able to send types of information, ultimately intended for human consumption, not expressable in plain ASCII. Thus, there is no way in plain ASCII to send the italics, boldface, or greek characters that can be expressed in Postscript. If some different processing is necessary to render these glyphs, then that is the minimum price to be paid in order to send them at all.

Second, by insisting that the message body contain only ASCII, we insure that it will not "break" current mail reading systems which are not equipped to process the structure; the result on the screen may not be readily interpretable by the human reader, however.

If a message sender knows that the recipient cannot process Postscript, he or she may prefer that the message be revised to eliminate the use of italics and boldface, rather than appear incomprehensible. If Postscript is being used because the message contains passages in Greek, there may be no suitable ASCII equivalent, however.

Ideally, the details of structuring the message (or not) to conform to the capabilities of the recipient system could be completely hidden from the message sender. The distributed Internet mail system would somehow determine the capabilities of the recipient system, and convert the message automatically; or, if there was no way to send Greek text in ASCII, inform the sender that his message could not be transmitted.

In practice, this is a difficult task. There are three possible approaches:

1. Each mail system maintains a database of capabilities of remote systems it knows how to send to. Such a database would be very difficult to keep up to date.

2. The mail transport service negotiates with the receiving system as to its capabilities. If the receiving system cannot support the specified content type, the mail is transformed into conventional ASCII before transmission. This would require changes to all existing SMTP implementations, and could not be implemented in the case where RFC-822 type messages are being forwarded via Bitnet or other networks which do not implement SMTP.

3. An expanded directory service maintains information on mail processing capabilities of receiving hosts. This eliminates the need for real-time negotiation with the final destination, but still requires direct interaction with the directory service. Since directory querying is part of mail sending as opposed to mail composing/reading systems, this requires changes to existing mailers as well as a major change to the domain name directory service.

We note in passing that the X.400 protocol implements approach number 2, and that the Draft Recommendations for X.DS, the Directory Service, would support option 3.

In the interest of facilitating early usage of structured messages, we choose not to recommend any of the three approaches described above at the present time. In a forthcoming RFC we will propose a solution based on option 2, requiring modification to mailers to support negotiation over capabilities. For the present, then, users would be obliged to keep their own private list of capabilities of recipients and to take care that they do not send Postscript, TROFF or other structured messages to recipients who cannot process them. The penalty for failure to do so will be the frustration of the recipient in trying to read a raw Postscript or TROFF file painted on his or her screen. Some System Administrators may attempt to implement option 1 for the benefit of their users, but this does not impose a requirement for changes on any other mail system.

We recognize that the long-term solution must require changes to mailers. However, in order to begin now to standardize the header fields, and to facilitate experimentation, we issue the present RFC.

3. The Content-type Header Field

Whatever structuring technique is specified by the Content-type field, it must be known precisely to both the sender and the recipient of the message in order for the message to be properly interpreted. In general, this means that the allowed parameter values for the Content-type: field must identify a well-defined, standardized, document structuring technique. We do not preclude, however, the use of a Content-type: parameter value to specify a private structuring technique known only to the sender and the recipient.

More precisely, we propose that the Content-type: header field consist of up to four parameter values. The first, or type parameter names the structuring technique; the second, optional, parameter is a version number, ver-num, which indicates a particular version or revision of the standardized structuring technique. The third parameter is a resource reference, resource-ref, which may indicate a standard database of information to be used in interpreting the structured document. The last parameter is a comment.

In the Extended Backus Naur Form of RFC-822, we have:

```
Content-Type:= type [";" ver-num [";" 1#resource-ref]] [comment]
```

3.1. Type Values

Initially, the type parameter would be limited to the following set of values:

```
type:=              "POSTSCRIPT"/"SCRIBE"/"SGML"/"TEX"/"TROFF"/
                    "DVI"/"X-"atom
```

These values are not case sensitive. POSTSCRIPT, Postscript, and POStscriPT are all equivalent.

POSTSCRIPT	Indicates the enclosed document consists of information encoded using the Postscript Page Definition Language developed by Adobe Systems, Inc. [1]
SCRIBE	Indicates the document contains embedded formatting information according to the syntax used by the Scribe document formatting language distributed by the Unilogic Corporation. [6]
SGML	Indicates the document contains structuring information to according the rules specified for

the Standard Generalized Markup Language, IS 8879, as published by the International Organization for Standardization. [3] Documents structured according to the ISO DIS 8613--Office Docment Architecture and Interchange Format--may also be encoded using SGML syntax.

TEX — Indicates the document contains embedded formatting information according to the syntax of the TEX document production language. [4]

TROFF — Indicates the document contains embedded formatting information according to the syntax specified for the TROFF formatting package developed by AT&T Bell Laboratories. [5]

DVI — Indicates the document contains information according to the device independent file format produced by TROFF or TEX.

"X-"atom — Any type value beginning with the characters "X-" is a private value.

3.2. Version Number

Since standard structuring techniques in fact evolve over time, we leave room for specifying a version number for the content type. Valid values will depend upon the type parameter.

ver-num:= local-part

In particular, we have the following valid values:

For type=POSTSCRIPT

ver-num:= "1.0"/"2.0"/"null"

For type=SCRIBE

ver-num:= "3"/"4"/"5"/"null"

For type=SGML

ver-num:="IS.8879.1986"/"null"

3.3. Resource Reference

resource-ref:= local-part

As Apple has demonstrated with their implementation of the Laserwriter, a very general document structuring technique can be made more efficient by defining a set of macros or other similar resources to be used in interpreting any transmitted stream. The Macintosh transmits a LaserPrep file to the Laserwriter containing font and macro definitions which can be called upon by subsequent documents. The result is that documents as sent to the Laserwriter are considerably more compact than if they had to include the LaserPrep file each time. The Resource Reference parameter allows specification of a well known resource, such as a LaserPrep file, which should be used by the receiving system when processing the message.

Resource references could also include macro packages for use with TEX or references to preprocessors such as eqn and tbl for use with troff. Allowed values will vary according to the type parameter.

In particular, we propose the following values:

For type = POSTSCRIPT

```
resource-ref:=  "laserprep2.9"/"laserprep3.0"/"laserprep3.1"/
                "laserprep4.0"/local-part
```

For type = TROFF

```
resource-ref:=  "eqn"/"tbl"/"me"/local-part
```

3.4. Comment

The comment field can be any additional comment text the user desires. Comments are enclosed in parentheses as specified in RFC-822.

4. Conclusion

A standardized Content-type field allows mail reading systems to automatically identify the type of a structured message body and to process it for display accordingly. The strcutured message body must still conform to the RFC-822 requirements concerning allowable characters. A mail reading system need not take any specific action upon receiving a message with valid Content-Type header field. The ability to recognize this field and invoke the appropriate display process accordingly will, however, improve the readability of messages, and allow the exchange of messages containing mathematical symbols, or foreign language characters.

In the near term, the major use of a Content-Type: header field is likely to be for designating the message body as containing a Page Definition Language representation such as Postscript.

Additional type values shall be registered with Internet Assigned Numbers Coordinator at USC-ISI. Please contact:

Joyce K. Reynolds
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292-6695

213-822-1511 JKReynolds@ISI.EDU

REFERENCES

1. Adobe Systems, Inc. Postscript Language Reference Manual. Addison-Wesley, Reading, Mass., 1985.

2. Crocker, David H. RFC-822: Standard for the Format of ARPA Internet Text Messages. Network Information Center, August 13, 1982.

3. ISO TC97/SC18. Standard Generalized Markup Language. Tech. Rept. DIS 8879, ISO, 1986.

4. Knuth, Donald E. The TEXbook. Addison-Wesley, Reading, Mass., 1984.

5. Ossanna, Joseph F. NROFF/TROFF User's Manual. Bell Laboratories, Murray Hill, New Jersey, 1976. Computing Science Technical Report No.54.

6. Unilogic. SCRIBE Document Production Software. Unilogic, 1985. Fourth Edition.